(12) UK Patent Application (19) GB (11) 2 267 193 (13) A

(43) Date of A publication 24.11.1993

(21) Application No 9210861.2

(22) Date of filing 21.05.1992

(51) INT CL[5]
H03H 17/06

(52) UK CL (Edition L)
H3U UGF

(56) Documents cited
US 4472785 A

(58) Field of search
UK CL (Edition K) H3U UGF
INT CL[5] H03H 17/06
Online databases: WPI

(71) Applicant
Sony United Kingdom Limited

(Incorporated in the United Kingdom)

Sony House, South Street, Staines, Middlesex, TW18 4PF, United Kingdom

(72) Inventors
Ahmed Sadjadian
Terence Ralph Hurley

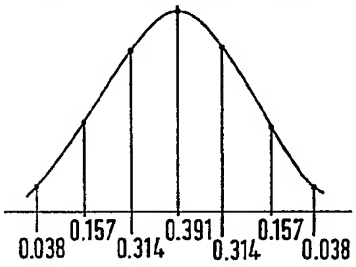
(74) Agent and/or Address for Service
D Young & Co
21 New Fetter Lane, London, EC4A 1DA, United Kingdom

(54) Multi-tap oversampling filter

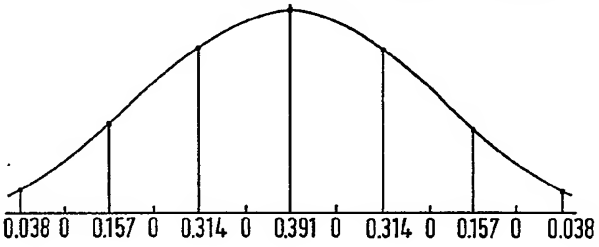
(57) A multi-tap oversampling filter for oversampling input data by an oversampling factor of m × n (where m and n are integers) is described. The filter coefficients of the multi-tap oversampling filter 9(c) are defined by a convolution of the coefficients of a first oversampling filter 9(a) having an oversampling factor of m with the coefficients of a second oversampling filter 9(b), where the second oversampling filter has an oversampling factor of n and has m–1 fixed-value coefficients (preferably zeroes) inserted between adjacent filter coefficients.

Fig.9

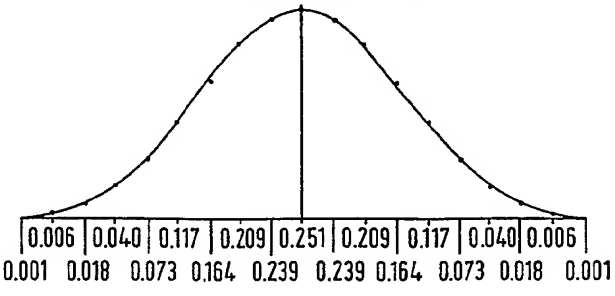
(a) x2 Oversampling filter.



(b) x2 Oversampling filter, with zeroes inserted.



(c) x4 Oversampling filter.



GB 2 267 193 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

The print reflects an assignment of the application under the provisions of Section 30 of the Patents Act 1977.

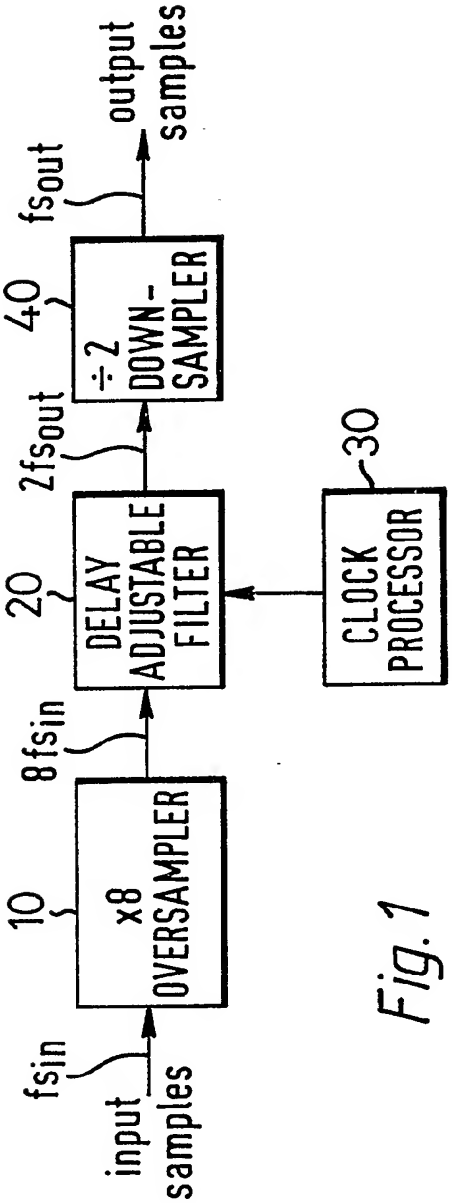

input samples
fsin
10
x8 OVERSAMPLER
8fsin
20
DELAY ADJUSTABLE FILTER
2fsout
40
÷2 DOWN-SAMPLER
fsout
output samples
30
CLOCK PROCESSOR

Fig. 1

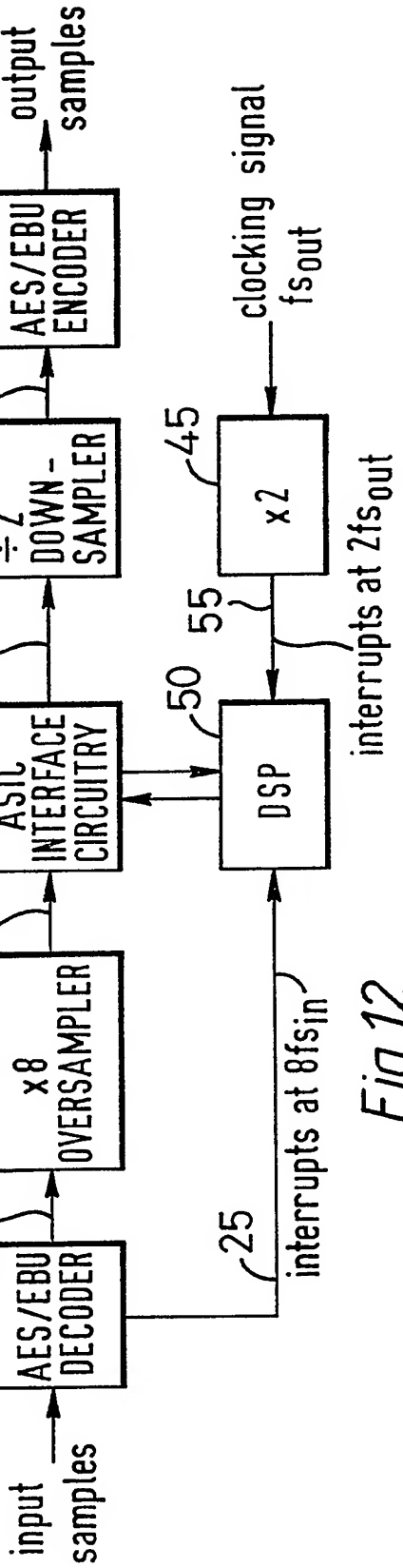

input samples
15
AES/EBU DECODER
fsin
10
x8 OVERSAMPLER
8fsin
60
ASIC INTERFACE CIRCUITRY
2fsout
40
÷2 DOWN-SAMPLER
fsout
35
AES/EBU ENCODER
output samples
25
interrupts at 8fsin
50
DSP
55
interrupts at 2fsout
45
x2
clocking signal fsout

Fig. 12

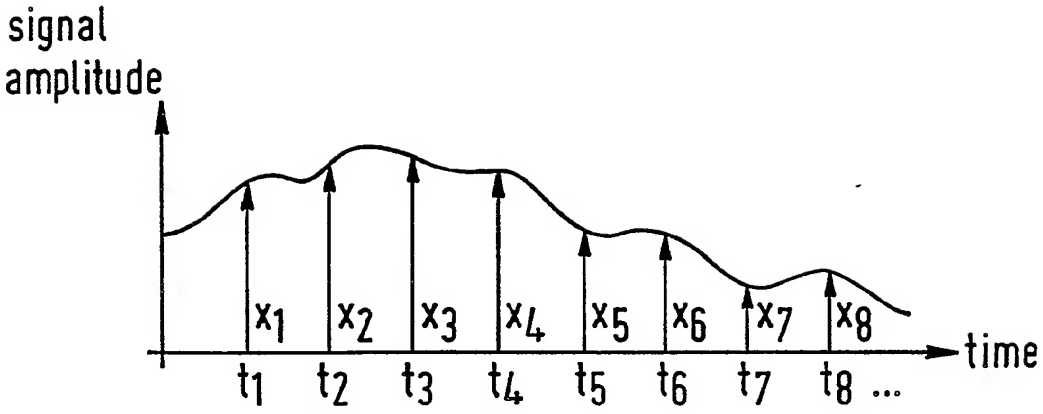


Fig. 2

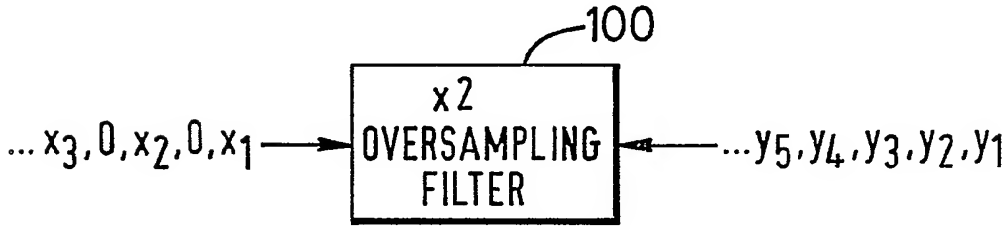


Fig. 3

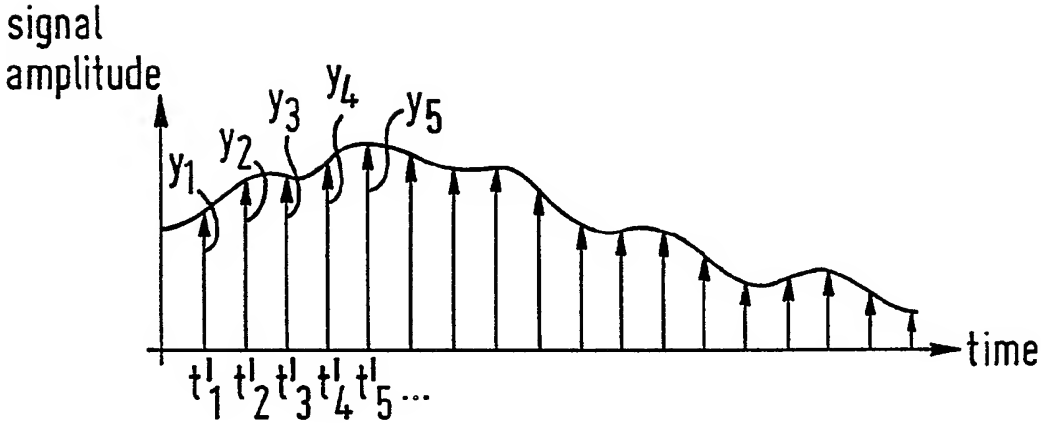


Fig. 4

Fig.5

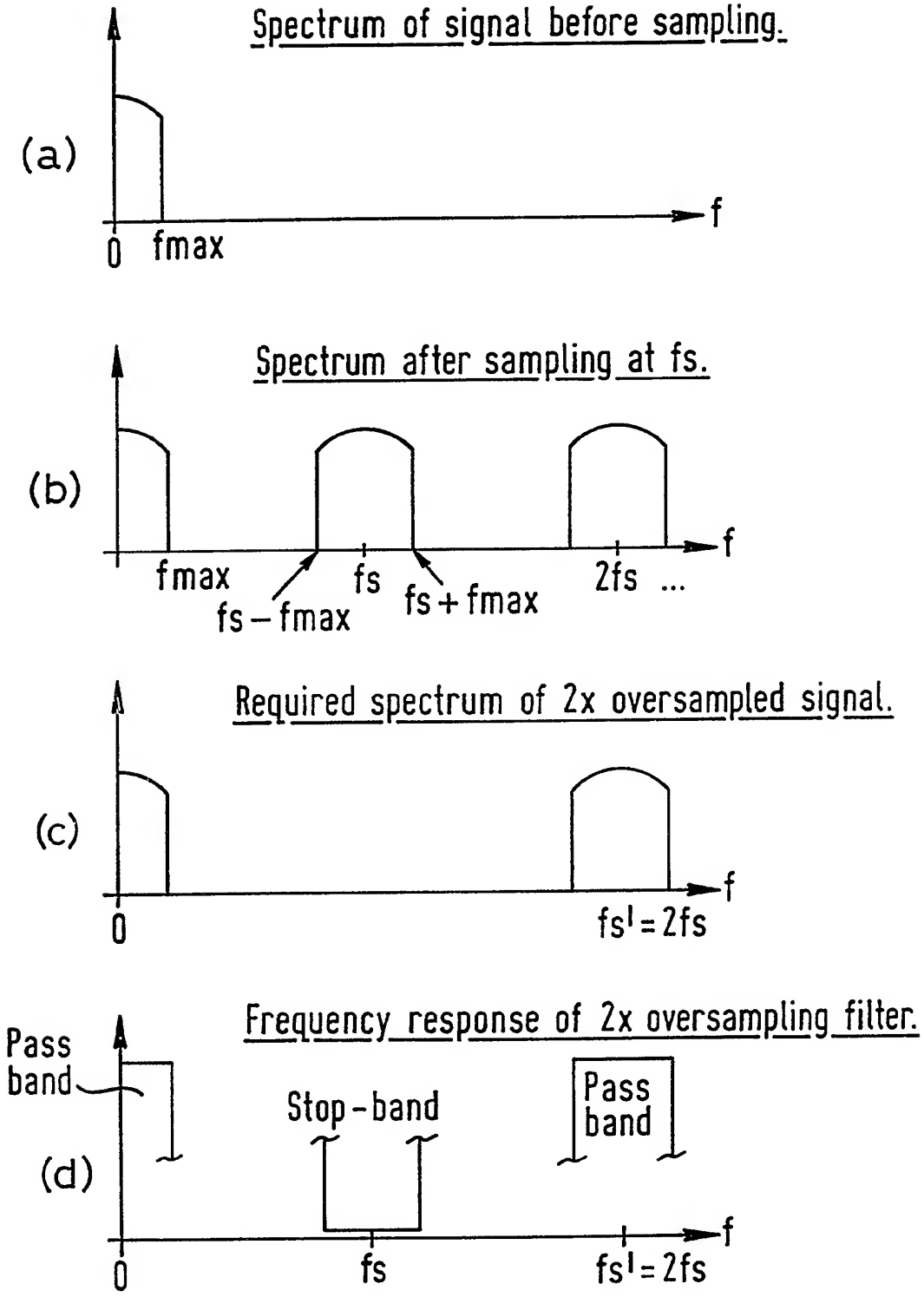
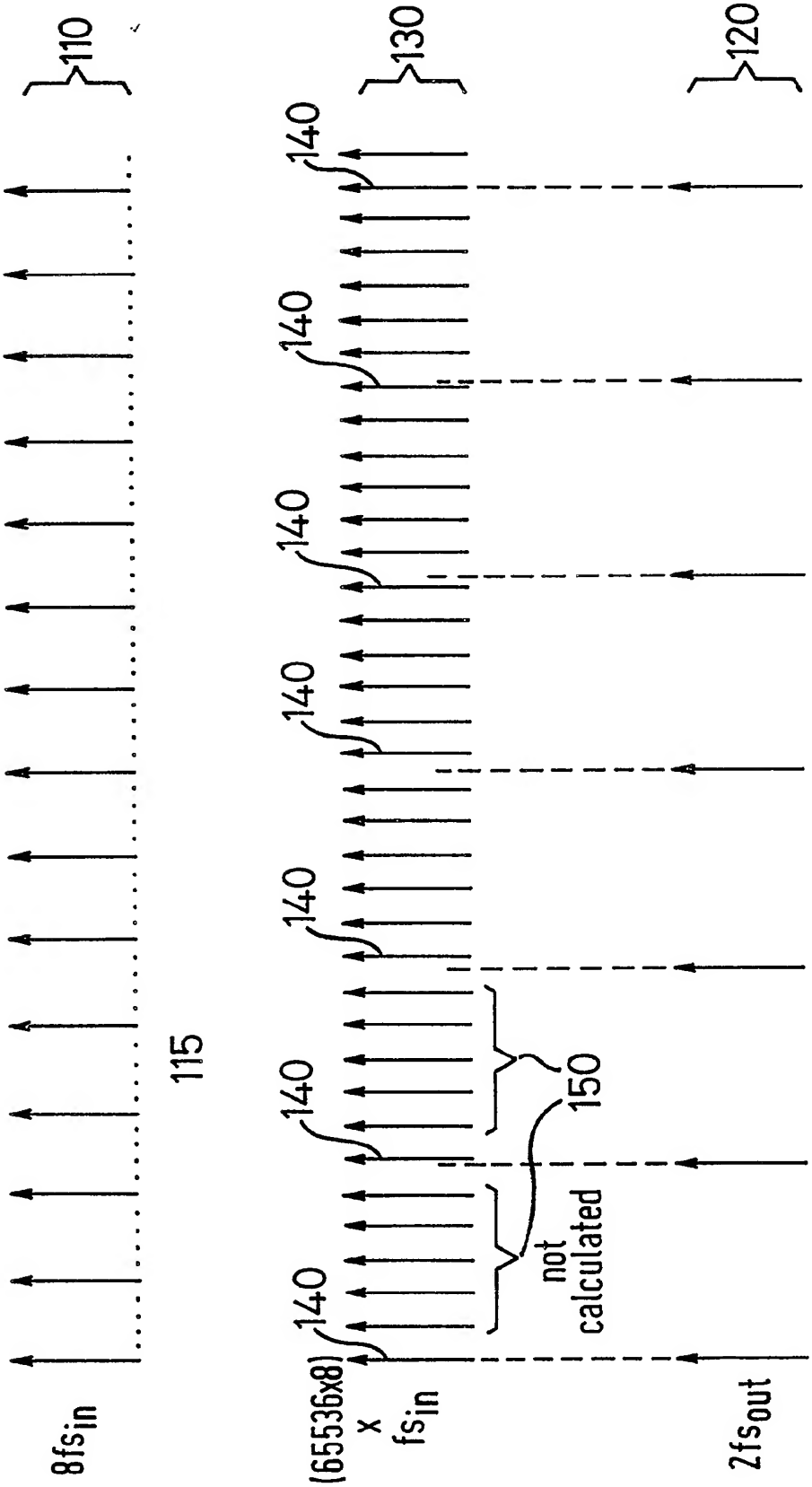

Spectrum of signal before sampling.
(a)
f
0
fmax
Spectrum after sampling at fs.
(b)
f
fmax
fs – fmax
fs
fs + fmax
2fs ...
Required spectrum of 2x oversampled signal.
(c)
f
0
fs' = 2fs
Frequency response of 2x oversampling filter.
Pass band
(d)
Stop-band
Pass band
f
0
fs
fs' = 2fs

Fig.6


8fs$_{in}$
110
115
(65536x8)
x
fs$_{in}$
140
not
calculated
150
130
120
2fs$_{out}$

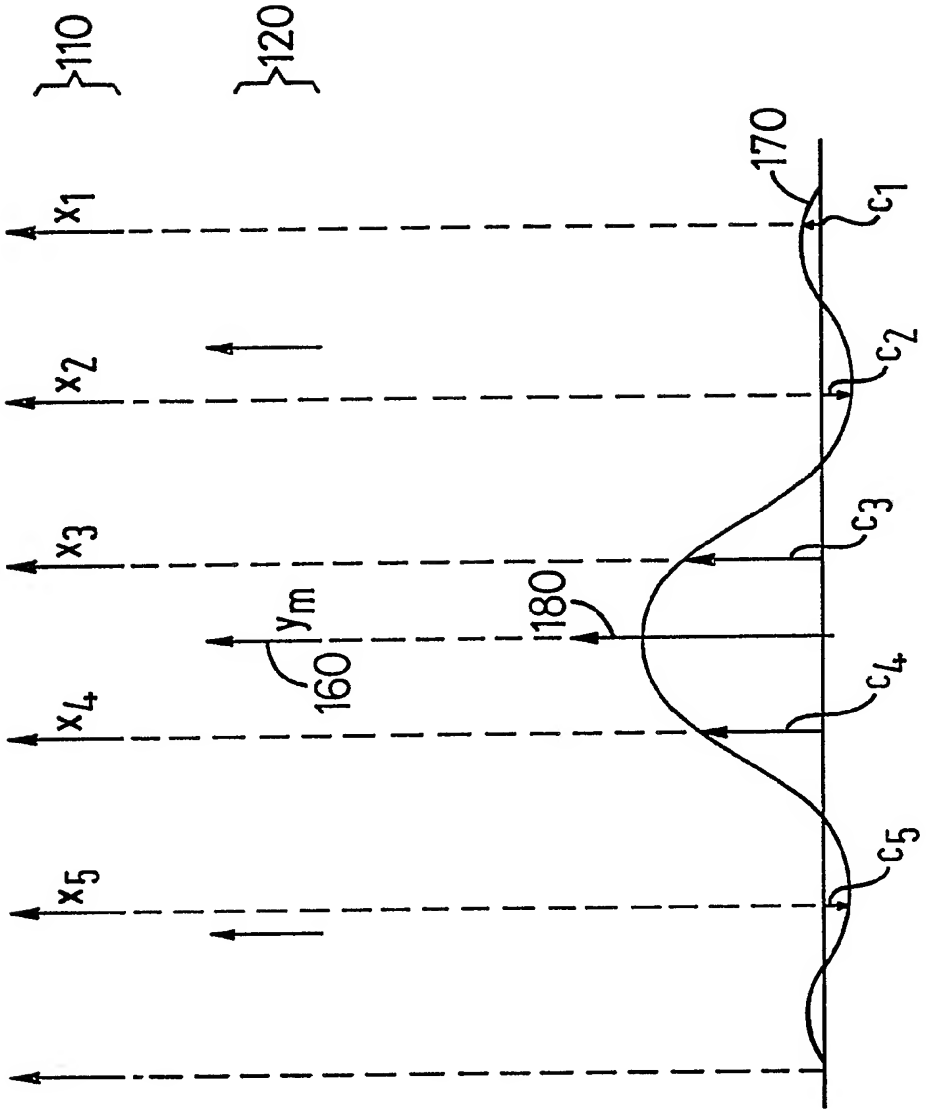

110
120
x1
x2
x3
x4
x5
ym
160
180
170
c1
c2
c3
c4
c5
8fsin
2fsout
yn = c1x1 + c2x2 + c3x3 + c4x4 + c5x5

Fig.7

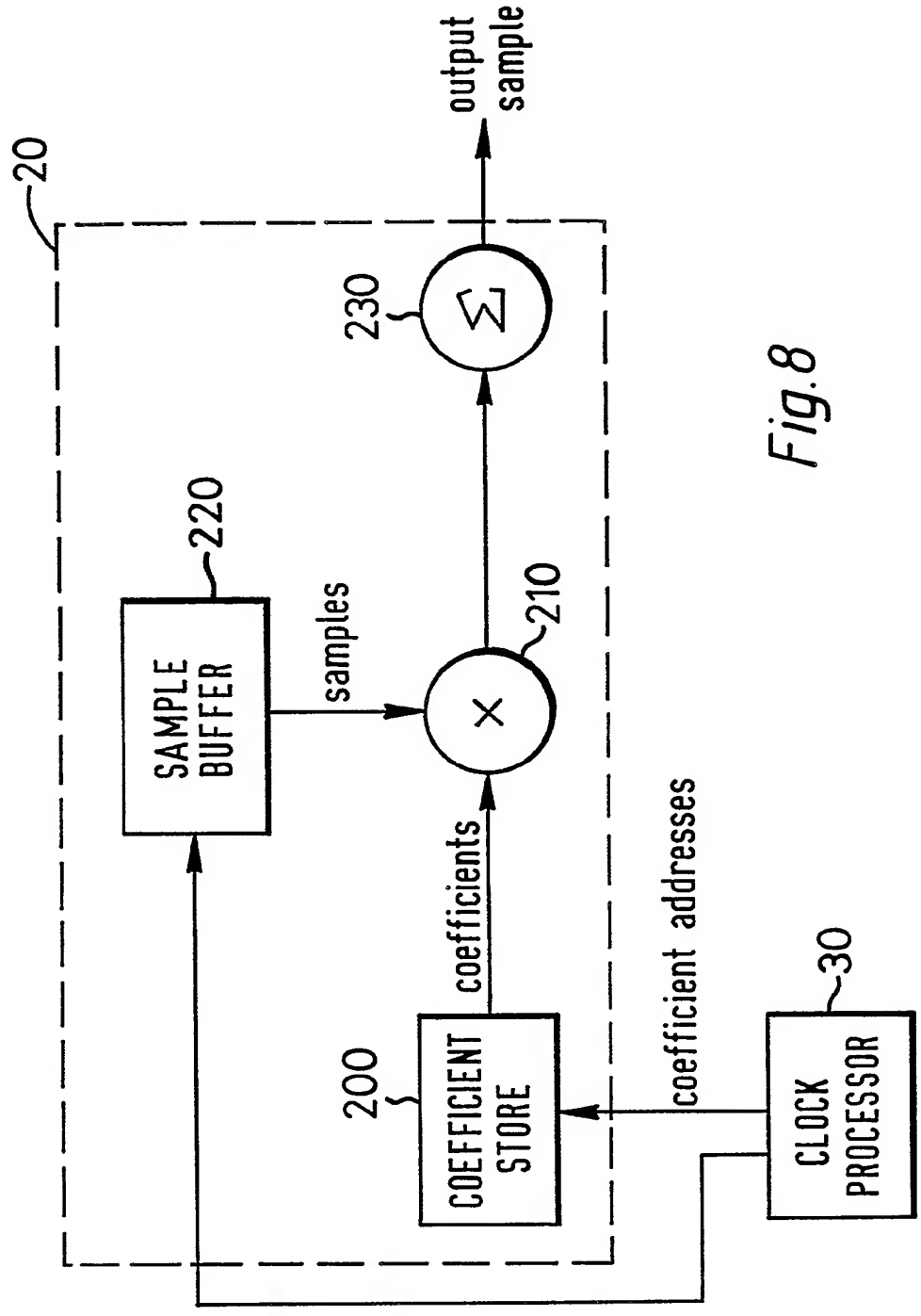
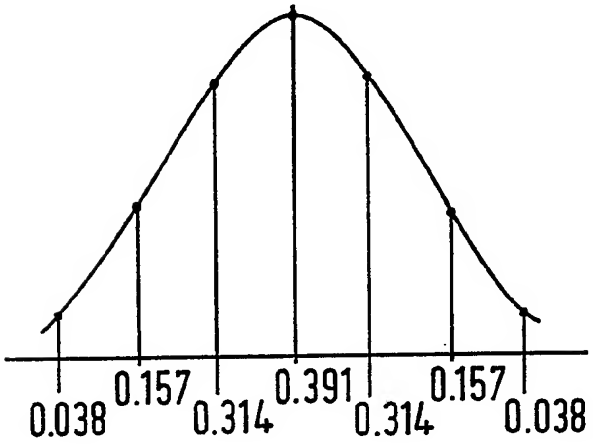

20
output sample
230
Σ
220
SAMPLE BUFFER
samples
210
×
coefficients
coefficient addresses
200
COEFFICIENT STORE
CLOCK PROCESSOR
30

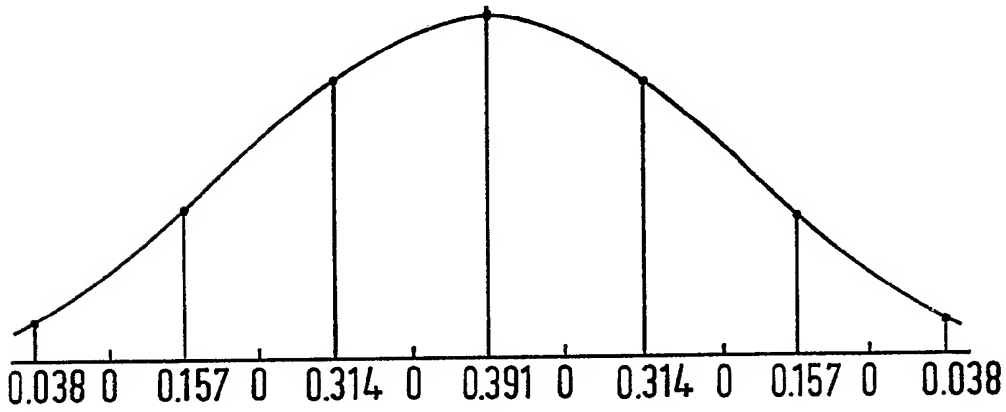
Fig.8

Fig.9

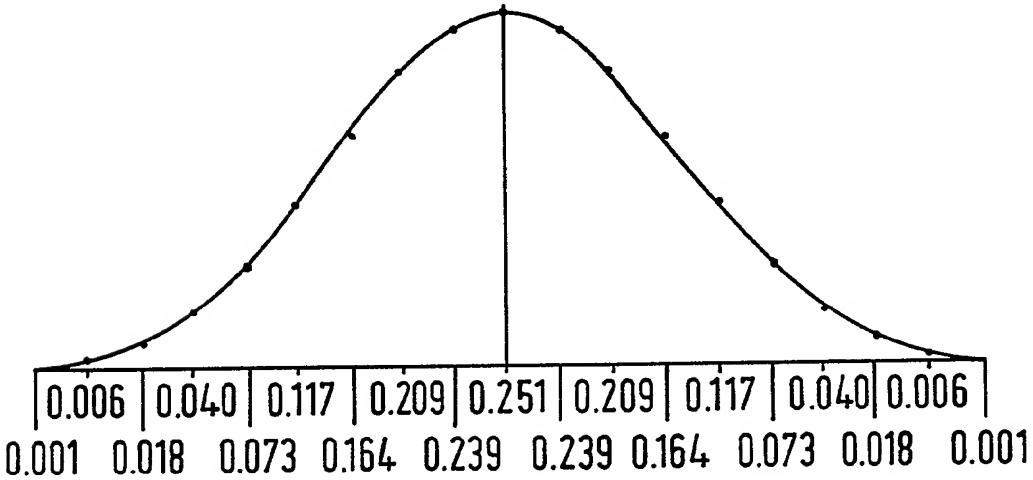
(a) x2 Oversampling filter.



(b) x2 Oversampling filter, with zeroes inserted.



(c) x4 Oversampling filter.



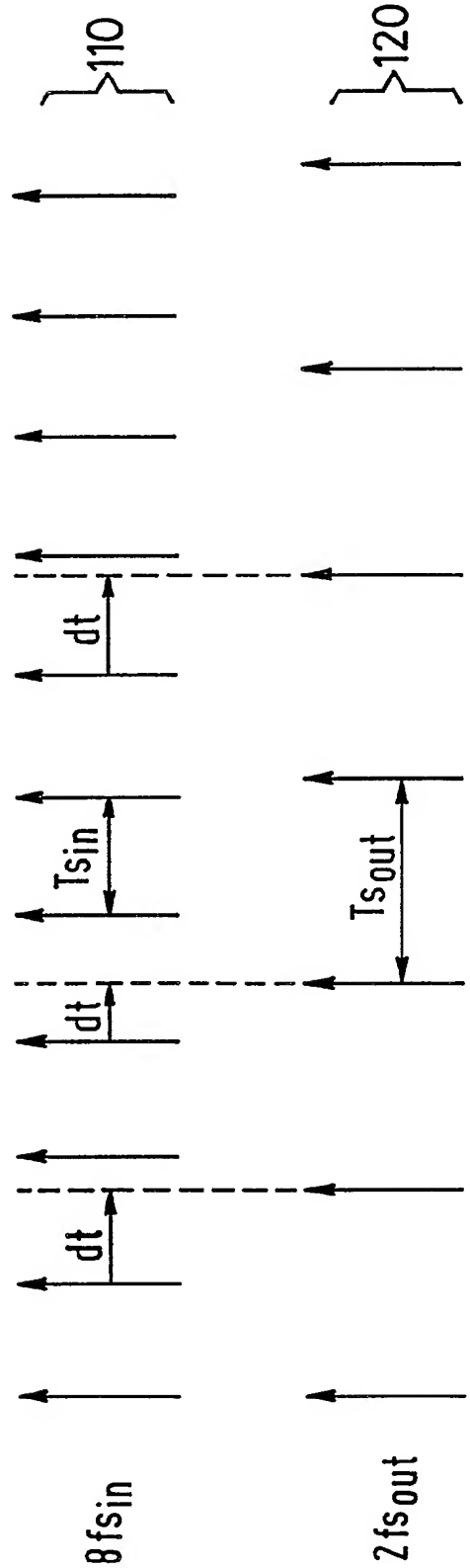


Fig. 10

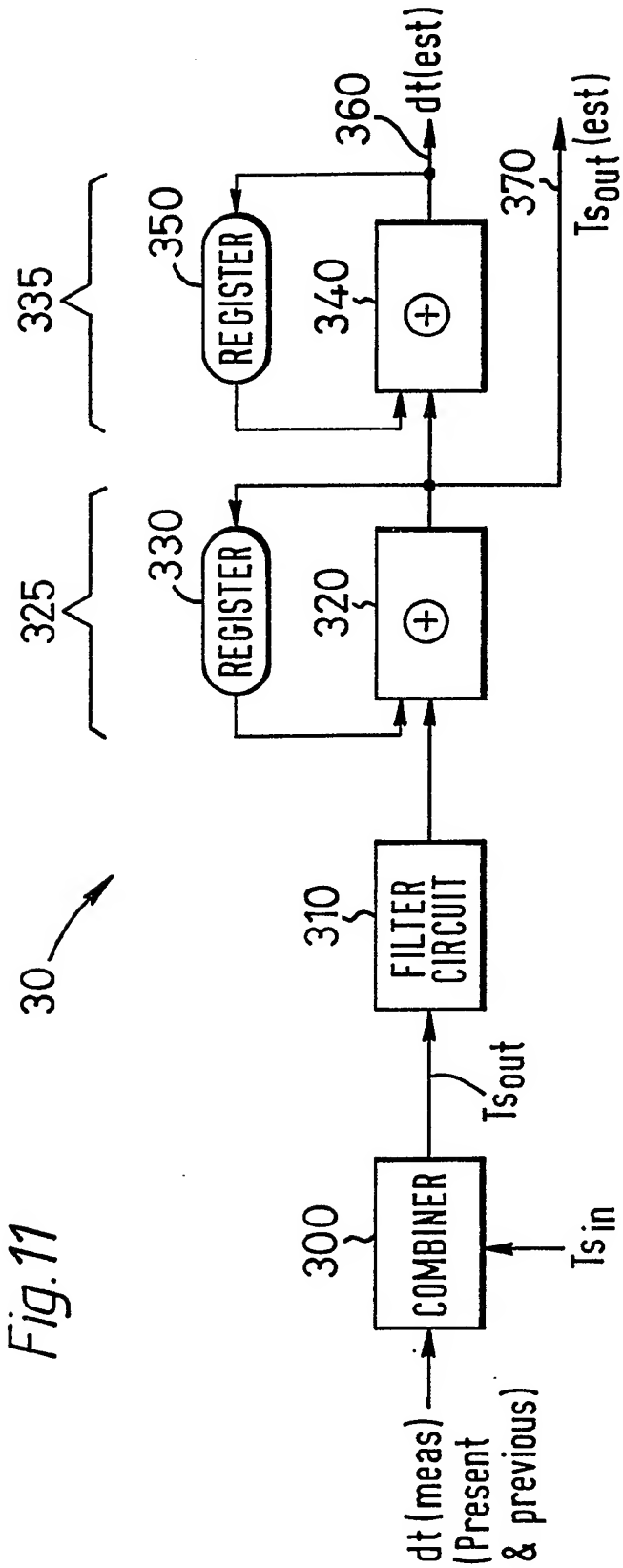

30
300
COMBINER
dt (meas)
(Present
& previous)
Ts in
Ts out
310
FILTER
CIRCUIT
325
330
REGISTER
320
335
350
REGISTER
340
360
dt(est)
370
Ts out (est)

Fig. 11

MULTI-TAP OVERSAMPLING FILTER

This invention relates to multi-tap oversampling filters.

Multi-tap oversampling filters operate on an input signal sampled at a particular sampling frequency (fs), to generate an output signal having a higher sampling frequency. In these filters, zero-valued samples are inserted between successive samples of the input signal to increase the number of samples in the input signal to that in the required output signal. After zero-insertion the input samples are then subjected to a filtering process in which each sample of the output signal is derived from a sum of the products of a group of adjacent samples and corresponding filter coefficients of the oversampling filter.

The ratio of the sampling frequency of the output signal to that of the input signal before zero-insertion will be referred to herein as the oversampling factor of the filter. The required frequency response of the filter depends on the oversampling factor: for an oversampling factor of I, an oversampling filter should have a single pass-band extending between the frequencies of zero and fs/2 (where fs/2 is one half of sampling frequency fs of the input signal before zero-insertion), (I-1) stop-bands centred around multiples of fs from 1fs to (I-1)fs, and a further passband centred around the frequency Ifs. The filter coefficients used in an oversampling filter are selected so that the frequency response of the filter matches, or at least approaches, this required response.

For small oversampling factors, in a range from 2 to perhaps 100, the filter coefficients can be derived using the Remez Exchange Algorithm, described in the book 'Theory and Practice of Digital Signal Processing' (Rabiner and Gold, Prentice Hall, 1975). However, the time taken to derive a set of coefficients using this method increases dramatically as the oversampling factor increases. For example, using present day computing apparatus it could take many years to derive a set of coefficients for use in a filter with an oversampling factor of, say, 65536.

This invention provides a multi-tap oversampling filter for oversampling input data by an oversampling factor of m x n, m and n being integers, the filter coefficients of the multi-tap oversampling

filter being defined by a convolution of the coefficients of a first oversampling filter having an oversampling factor of m with the coefficients of a second oversampling filter, the second oversampling filter having an oversampling factor of n and having m-1 fixed-value coefficients inserted between adjacent filter coefficients.

An oversampling filter according to the invention has coefficients which are defined by a combination of the coefficients of at least two other filters having oversampling factors lower than that of the required filter. This means that the coefficients of a filter having a high oversampling factor, which would require a large amount of computation to derive, can be generated from those of filters having sufficiently low oversampling factors that their coefficients can be derived using techniques such as the Remez exchange algorithm.

Preferably, in order to simplify the required calculations, the fixed-value coefficients are zero-valued coefficients.

The definition of the coefficients of the multi-tap oversampling filter can be a multi-stage process. In particular, the filter coefficients of the first or the second multi-tap oversampling filter could be generated by zero-insertion and convolution of two other multi-tap oversampling filters, which could in turn be generated from other, smaller, filters.

Viewed from a second aspect this invention provides a sampling frequency converter for operating on sampled input data having an input sampling frequency to generate sampled output data having an output sampling frequency, the sampling frequency converter comprising: a multi-tap oversampling filter as described above for operating on the input data to generate oversampled input data, the oversampled input data having a sampling frequency greater than the output sampling frequency; and a downsampler for downsampling the oversampled input data to generate the output data.

In a preferred embodiment the downsampler comprises means for selecting a respective sample of the oversampled input data for use as each sample of the output data, the selected sample of the oversampled input data being the sample temporally closest to that sample of the output data. Although the oversampled input samples not selected by the downsampler could be generated and then immediately discarded, it is preferable in order to reduce the complexity and the required

operation speed of the sampling frequency converter that samples of the oversampled input data not selected by the downsampler are not generated.

Viewed from a third aspect this invention provides a method of generating filter coefficients for a multi-tap oversampling filter for oversampling input data by an oversampling factor of m x n, m and n being integers, the method comprising the steps of: inserting m-1 fixed-value coefficients between adjacent filter coefficients of a first oversampling filter having an oversampling factor of n; and convolving the coefficients of the first oversampling filter with the coefficients of a second oversampling filter having an oversampling factor of m.

Viewed from a fourth aspect this invention provides a method of increasing the oversampling factor of a multi-tap oversampling filter by a factor of n, where n is an integer, the method comprising the steps of: inserting n-1 fixed-value coefficients between adjacent filter coefficients of the multi-tap oversampling filter; and convolving the coefficients of the multi-tap oversampling filter with the coefficients of a second multi-tap oversampling filter having an oversampling factor of n.

Again, in the second and third aspects of the invention, it is preferred that the fixed-value coefficients are zero-valued coefficients.

An embodiment of the invention will now be described by way of example only with reference to the accompanying drawings, throughout which like parts are referred to by like references, and in which:

Figure 1 is a schematic block diagram showing the function of a digital audio sample rate converter;

Figure 2 illustrates a sampled analogue signal in the time domain;

Figure 3 is a schematic diagram showing the operation of an oversampling filter;

Figure 4 illustrates an oversampled version of the sampled analogue signal of Figure 12;

Figure 5 is a schematic illustration showing the oversampling process in the frequency domain;

Figure 6 illustrates the operation of a delay adjuster filter;

Figure 7 illustrates the selection of filter coefficients in the delay adjuster filter of Figure 6;

Figure 8 is a schematic block diagram of the delay adjuster filter of Figure 6;

Figure 9 illustrates the derivation of filter coefficients for an oversampling filter;

Figure 10 illustrates the timing signals used in a clock processor;

Figure 11 is a schematic block diagram of a clock processor; and

Figure 12 is a schematic diagram showing an implementation of the digital audio sampling rate converter of Figure 1.

Referring now to Figure 1, a schematic block diagram is shown which illustrates the function of a digital audio sampling rate converter (DASRC). An input audio signal, sampled at a sampling rate of fs_{in}, is passed first to an eight-times oversampler 10 which converts the input signal sampled at fs_{in} into an oversampled input signal having a sampling frequency of $8fs_{in}$. The oversampled input signal is passed to a delay adjuster filter 20 which, under the control of a clock processor 30, converts the oversampled input data to oversampled output data at a sampling rate of $2fs_{out}$ (twice the desired output sampling rate). The oversampled output signal is then passed to a two-times downsampler 40 which generates an output signal sampled at an output sampling rate of fs_{out}.

Figure 2 is an illustration in the time domain of a sampling process applied to an analogue signal. Figure 2 is a graph showing the amplitude of the analogue signal on a vertical axis, against time on a horizontal axis. The analogue signal is sampled at a number of temporally spaced instants t_1, t_2, t_3 ... to yield corresponding sample values x_1, x_2, x_3 ... . The sampling frequency is the reciprocal of the time period between successive sampling instants.

Figure 3 is a schematic diagram showing the operation of an oversampling filter on the sampled analogue signal of Figure 2. A two-times oversampling filter 100 is shown, which operates on successive input samples x_n to generate successive output signals y_n at twice the sampling frequency of the input samples. Before being processed by the oversampling filter 100 the input samples x_n are interspersed with zero-valued samples in order that the number of input samples applied to the

oversampling filter 100 is the same as the number of required output samples. In general, for a oversampling factor of I, (I-1) zero-valued samples must be placed between adjacent samples of the input signal. The output samples y_n represent the amplitude of the analogue signal at successive instants t_1', t_2', t_3' ... and are shown graphically in Figure 4 in relation to the original analogue signal.

Figure 5(a) to 5(d) are schematic illustrations, in the frequency domain, of the oversampling process performed by the oversampling filter 100 of Figure 3. In these illustrations, the signal amplitude is represented on a vertical axis, against signal frequency on a horizontal axis. Figure 5(a) illustrates the frequency spectrum of an example analogue signal before sampling is performed. Before any sampling takes place, the analogue signal occupies a frequency range from 0 to fmax.

Figure 5(b) illustrates the spectrum of a signal after a sampling process at a sampling frequency of fs has been performed. In addition to the original baseband signal (from 0 to fmax), sideband signals have been generated by the sampling process which are centred around harmonics of the sampling frequency fs. For example, a first sideband extends between the frequencies of (fs-fmax) and (fs+fmax). Similar sidebands are centred around 2fs, 3fs and so on.

Figure 5(c) illustrates the required spectrum of the signal of Figure 5(b) after that signal has been subjected to a two-times oversampling process. The effect of the oversampling process is to generate output samples (y_n in Figure 3) at a sampling frequency which is a multiple of the original sampling frequency of the signal. In the case of a two-times oversampling process, the sampling frequency (fs') of the output samples y_n will be twice that of the original sampled signal x_n. (That is to say, fs' = 2fs). This means that the two-times oversampled signal should have sidebands centred around fs'(=2fs), 2fs'(=4fs) and so on. A visual comparison of Figure 5(b) and Figure 5(c) demonstrates that the two-times oversampling filter 100 is required to remove the sidebands centred around <u>odd multiples</u> of the frequency fs.

The actual frequency response of the two-times oversampling filter is illustrated in Figure 5(d). As shown in Figure 5(d), the two-times oversampling filter 100 requires a passband ranging from 0 to

fmax, a stopband of width 2fmax centred around the frequency fs, and a further pass band of width 2fmax centred around the frequency fs' (where fs' = 2fs). In general, an oversampling filter having an oversampling factor of I requires a pass band ranging from 0 to at least fmax, (I-1) stop bands having a width of at least 2fmax centred around multiples of fs from 1fs to (I-1)fs, and a further pass band having a width of at least 2fmax centred around the frequency Ifs.

Figure 6 illustrates the operation of the delay adjuster filter 20 of Figure 1. It will be recalled that the samples supplied as an input to the delay adjuster filter 20 have a sampling frequency of $8fs_{in}$, since they have already undergone an eight-times oversampling process by the eight-times oversampler 10. Also, it will be recalled that the output samples generated by the delay adjuster filter 20 are at twice the required output sampling frequency (that is, $2fs_{out}$). In Figure 6, the input samples at the sampling frequency $8fs_{in}$ supplied to the delay adjuster filter 20 are shown schematically at respective positions 110 along a horizontal time axis, and the output samples at the sampling frequency of $2fs_{out}$ are similarly shown at respective positions 120 along the horizontal time axis.

The delay adjuster filter 20 operates basically to oversample the input samples supplied to it by a very large oversampling factor, to generate intermediate samples 130 having a very small temporal separation. For each of the output samples 120, that one of the intermediate samples 130 having a temporal position closest to that of the output sample 120 is used as that output sample 120.

In this embodiment, the oversampling factor used in the delay adjuster filter 20 is 65536, although (for clarity of the diagram) such a large oversampling factor has not been shown in Figure 6. In theory, in order to perform a process of 65536-times oversampling, 65535 zero-valued samples must be inserted between each of the input samples supplied to the delay adjuster filter 20. In practice, this requirement is avoided, but for the purposes of explaining Figure 6 it will be assumed initially that zero-insertion has been performed. The zero-valued samples are shown schematically in Figure 6 as points (e.g. the point 115) between adjacent input samples 110.

It would be possible for the delay adjuster filter 20 to calculate all 65536 intermediate samples 130 for each input sample 110,

to use selected ones of the intermediate samples 130 as two-times oversampled output samples 120 and to discard the remaining vast majority of the intermediate samples 130. However, since 65535 out of every 65536 intermediate samples are discarded, a considerable saving in computation time can be achieved by calculating only the relevant one out of every 65536 intermediate samples. In practice therefore, the delay adjuster filter 20 operates in conjunction with the clock processor 30 to select, for each output sample at the sampling frequency of $2fs_{out}$, that one 140 of the 65536-times oversampled samples which is nearest to the temporal position of the required output sample. This selected oversampled sample 140 is then calculated and output as an output sample. The intervening oversampled samples 150 are not calculated.

As mentioned above, for each output sample at $2fs_{out}$, the nearest oversampled sample 140 is selected and used as that output sample 120. For some of the output samples 120, the temporal position of the selected oversampled sample 140 may not coincide exactly with the temporal position of the required output sample. This will generally be the case when the output sampling frequency fs_{out} is not related by a simple ratio to the input frequency fs_{in}. However, neglecting any errors in the calculation of the temporal position of the output samples, the maximum error in temporal position which can occur is one half of the temporal separation of the 65536-times oversampled samples 130. In the present embodiment, for an input sampling frequency fs_{in} of 48kHz, the temporal separation of successive 65536-times oversampled samples 130 is approximately 40 picoseconds (pS) so the maximum temporal error in an output sample is approximately 20 pS.

The 65536-times oversampling filter used in the delay adjuster filter 20 has a very large number (323307) of coefficients, the derivation of which will be described below with reference to Figure 9. In principle, when one of the samples 140 is calculated, each of these coefficients is multiplied by the value of a corresponding zero-inserted input sample 110 (assuming that zero-insertion had taken place). However, the vast majority (65535 in every 65536) of the zero-inserted input samples 110 are zero-valued samples, so although these zero-valued samples could be multiplied by corresponding coefficients, the result of the multiplication would automatically be zero. In

practice therefore neither zero insertion nor the step of multiplying coefficients by zero is actually performed.

As mentioned above, the 65536-times oversampling filter used in the delay adjuster filter 20 in this embodiment has 323307 coefficients, each of which is (in theory) multiplied by a corresponding one of the zero-inserted input samples 110. However, since zero-insertion is not performed, only one in 65536 of the input samples 110 to be multiplied by corresponding coefficients will be present. This means that the number of actual multiplications which needs to be performed to generate one output sample 140 is equal to:

$$\textit{integer value of } \left(\frac{323307}{65536} + 1\right) = 5 \textit{ multiplications}$$

The way in which five coefficients are selected from the 323307 possible coefficients for use in generation of each output sample will be described below with reference to Figure 7.

Figure 7 is a schematic illustration showing the selection of the five filter coefficients used in the derivation of each output sample. In particular, the coefficient for use in the generation of an output sample y_m 160 will be shown. In Figure 7 the 323307 individual coefficients in the 65536-times oversampling filter used in the delay adjuster filter 20 are not shown individually, for reasons of clarity of the diagram, but instead their envelope is shown by a curve 170.

In order to derive the output sample 160, the centre coefficient 180 of the set of 323307 coefficients is aligned with the temporal position of the output sample 160 and the coefficients nearest to the temporal positions of the 5 input samples 110 surrounding that output sample 160 are selected for multiplication by the corresponding input sample. In the example shown in Figure 7, an input sample x_1 is multiplied by a filter coefficient c_1; an input sample x_2 is multiplied by a filter coefficient c_2; and so on up to an input sample x_5 which is multiplied by a filter coefficient c_5. The output sample is then generated from the sum of the products of the above multiplications so that:

Figure 8 is a schematic block diagram showing the delay adjuster filter 20 and its relationship to the clock processor 30.

$$y_n = C_1.X_1 + C_2.X_2 + C_3.X_3 + C_4.X_4 + C_5.X_5 = \sum_{i=1}^{i=5} C_i.X_i$$

The clock processor 30 is responsive to the temporal positions of the input samples (supplied to the delay adjuster filter 20) and of the output samples (to be generated by the delay adjuster filter 20) in order to generate coefficient addresses for accessing the appropriate coefficients in a coefficient store 200 in the manner described above with reference to Figure 7. The coefficient store 200 may be implemented as a programmable read only memory to store numerical representations of the coefficients. (Because the filter coefficients in the 65536-times oversampling filter are symmetrical about the central coefficient 180, only one half of the coefficients needs to be stored in the coefficient store 200. Also, rounding the coefficients to 24-bit precision for storage means that only 322345 out of the 323307 coefficients are non-zero).

In response to each of the five coefficient addresses supplied by the clock processor 30, the coefficient store 200 supplies a corresponding filter coefficient to a multiplier 210. At the same time, a sample buffer 220 storing the eight-times oversampled input samples 110 supplies an appropriate sample to the multiplier 210 to be multiplied by that filter coefficient. The products of the multiplications performed by the multiplier 210 are supplied to a summation unit 230 which adds the five products to generate an output sample. This five-stage process is repeated for each output sample to be generated by the delay adjuster filter 20.

It is not possible to use a known filter coefficient derivation algorithm such as the Remez Exchange algorithm to generate the coefficients in the 65536-times oversampling filter used in the delay adjuster filter 20. This is because the computation time required to generate the coefficients would be prohibitive for such a large oversampling factor. Instead, the filter coefficients are generated as follows:

(a) First an optimum multi-band digital filter interpolator capable of oversampling by an oversampling factor of I_1 is designed using (for example) the Remez Exchange algorithm. This filter is called the "base filter" and denoted hereinafter by F_1. The base filter

has two passbands (at zero frequency and at I_1 x the input sampling frequency) and (I_1-1) stopbands centred around harmonics of the input sampling frequency. The base filter has a relatively short length T_1 (in many cases less than 100 coefficients), so that the process of generating the coefficients of the base filter does not take a prohibitively long computation time, and that the final number of coefficients is kept low.

(b) The coefficients of the base filter F_1 are then combined with those of a second filter F_2, the second filter F_2 having T_2 coefficients and providing an oversampling factor of I_2. However, before this combination takes place the coefficients of the base filter are first zero inserted, which involves the insertion of (I_2 - 1) zero valued coefficients between each pair of adjacent coefficients of the base filter. (In fact any fixed value could be inserted in this way, but the use of zero simplifies the calculations involved in combining F_1 and F_2). If the base filter is originally of T_1 coefficients then after zero insertion the base filter will become a filter having $((T_1-1).\ I_2 + 1)$ coefficients.

After zero-insertion the coefficients of the base filter F_1 are then convolved with the coefficients of the second filter F_2. The length of the filter developed after convolution of the zero inserted filter with the second filter F_2 of length T_2 will be $T_1.I_2 + T_2 - I_2$. The resulting filter will have an oversampling factor of $I_1.I_2$. This filter is then further developed.

(c) The filter produced by combining the coefficients of F_1 and F_2 in step (b) is then zero inserted and convolved with a third oversampling filter F_3 which has T_3 taps and an oversampling factor of I_3. Here the zero insertion involves insertion of I_3-1 zero valued coefficients in between each pair of adjacent coefficients of the filter resulting from step (b). The filter produced by the convolution with F_3 has an oversampling factor of $I_1.I_2.I_3$ and has:

$T_1.I_2.I_3 + T_2.I_3 - I_2.I_3 - I_3 + T_3$ coefficients.

Step (c) is in effect a repetition of step (b) using F_3 instead of F_2, although the possibility of re-using one of the filters F_1 or F_2 is not excluded. Step (c) can be repeated as required using further

filters F_4, F_5... having T_4, T_5... coefficients and oversampling factors of I_4, I_5... respectively.

Assuming that the total interpolation factor required is realised using say, 3 stages of zero insertion followed by convolution (in other words, $I_{overall} = I_1.I_2.I_3.I_4$), the length of the resulting filter is equal to:

$$T_{overall} = T_1.I_2.I_3.I_4 + I_3.I_4.T_2 + T_3.I_4 + T_4 - (I_2.I_3.I_4 + I_3.I_4 + I_4)$$

The process described above can be generalised so that a filter having an oversampling factor of I_x can be zero inserted and convolved with another filter having an oversampling factor of I_y to generate the coefficients for a filter having an oversampling factor of $I_x.I_y$. In this way, filters having large oversampling factors can be constructed in stages, thus saving at least some of the computation time normally required to generate their coefficients.

Figure 9 is a schematic illustration of the above process in which a four-times oversampling filter (shown in Figure 9(c)) is generated by convolving the coefficients of a two-times oversampling filter shown in Figure 9(a) with the coefficients of a two-times oversampling filter with a zero inserted between each coefficient (shown in Figure 9(b)). The filter shown in Figure 9(a), and that shown in Figure 9(b) before zero insertion, have seven coefficients, namely;

0.038; 0.157; 0.314; 0.391; 0.314; 0.157; 0.038

These coefficients are listed beneath the horizontal axis in Figures 9(a) and 9(b). If the coefficients in the filter in Figure 9(a) are denoted by g_{-3} to g_3, and those in Figure 9(b) (i.e. after zero insertion) are denoted by h_{-6} to h_6, the convolution process generates each coefficient f_m of the output (ie. four-times oversampling) filter shown in Figure 9(c) as follows:

$$f_m = \sum_{i=-3}^{i=3} g_i.h_{m-i}$$

The four-times oversampling filter in Figure 9(c) is generated from a filter F_1 having seven coefficients and a filter F_2, also having seven coefficients. From step (b) above, the length of the resulting filter is $T_1.I_2+T_2-I_2$ = (7 x 2) + (7 - 2) = 19. The resulting filter

therefore has 19 coefficients, as shown in Figure 9(c).

The simple example shown in Figure 9 served to illustrate the filter coefficient derivation process described above. However, the process is particularly useful when filters with a large number of coefficients and/or a high oversampling factor are being generated. For example, in order to generate the coefficients of the 65536-times oversampling filter for use in the delay adjuster filter 20 of the present embodiment the following filters F_1 to F_4 are combined:

Filter	No. of Coefficients	Oversampling Factor
F_1	$T_1 = 75$	$I_1 = 16$
F_2	$T_2 = 75$	$I_2 = 16$
F_3	$T_3 = 75$	$I_3 = 16$
F_4	$T_4 = 75$	$I_4 = 16$
$F_{overall}$	$T_{overall} = 323307$	$I_{overall} = 65536$

The accuracy of generation of each output sample by the delay adjuster filter 20 depends on the accurate determination of the temporal position of that output sample with respect to the input samples, so that the appropriate coefficients in the 65536-times oversampling filter in the delay adjuster filter 20 can be selected. This determination of the temporal position is performed by the clock processor 30.

For an input sampling frequency (fs_{in}) of 48 Khz, the temporal resolution of the delay adjuster filter (that is, the temporal separation of adjacent intermediate samples 130) is about 40 pS. It is difficult to measure the temporal position of each output sample to this accuracy, because of jitter in the input and output clock frequencies and because such measurement would require an extremely fast timing circuit. However, satisfactory results can be obtained by measuring the position of each output sample to a certain accuracy (say, to the nearest 256 intermediate samples 130) and then performing a 'rolling average' of these measured values in order to generate a temporal position accurate to one intermediate sample.

Figure 10 is a schematic illustration showing some of the timing signals used by and generated by the clock processor 30, and Figure 11 is a schematic block diagram of the clock processor 30.

In Figure 10, the temporal separation of two successive eight-times oversampled output samples is referred to as Ts_{out}. This value is initially measured for a particular pair of eight-times oversampled input samples, (referred to as $Ts_{out}(meas)$) and then estimated ($Ts_{out}(est)$) by the clock processor to a higher precision than that of the measured value. The temporal separation of two successive eight-times oversampled input samples is referred to as Ts_{in}. Similarly dt is the time difference between the temporal position of each output sample and that of the immediately preceding eight-times oversampled input sample. Again, the value of dt is initially measured (dt(meas)) and then estimated (dt(est)) by the clock processor 30.

The clock processor 30 shown in Figure 11 comprises a combiner 300, a filter circuit 310, an adder 320 and a register 330 arranged as an averager 325, and a second adder 340 and a second register 350 arranged as a second averager 335. The clock processor supplies two outputs, namely a dt(est) output 360 and a $Ts_{out}(est)$ output 370.

The combiner receives values of Ts_{in} and the present and previous values of dt(meas), which are measured from the input and output clocking signals respectively. The combiner combines these values to generate values of $Ts_{out}(meas)$:

$$Ts_{out}(meas) = Ts_{in} - dt(meas)_{previous} + dt(meas)_{present}$$

The values of $Ts_{out}(meas)$ obtained in this way are smoothed by the filter circuit 310 and are then passed to the averager 325. The averager 325 serves to calculate a rolling average of the $Ts_{out}(meas)$ values, thereby generating a value of $Ts_{out}(est)$ which is supplied as the output 370.

The value of $Ts_{out}(est)$ generated by the averager 325 is also passed to the second averager 335, in which it is used to update the current value of dt(est):

$$dt(est)_{updated} = [\ dt(est)_{updated} + Ts_{out}(est)\ ] \text{ modulo } Ts_{in}$$

The updated value of dt(est) is supplied as the output 360.

The estimated value dt, dt(est), is used to determine the position of the centre coefficient 180 in the delay adjuster filter 20

with respect to the input samples. The positions of the remaining coefficients with respect to the centre coefficient 180 are found by simply adding multiples of 65536 to the memory address of the centre coefficient in the coefficient store 200, taking account of the fact that the filter is symmetrical and that only one half of the coefficients are stored.

The value of dt(est) computed by the clock processor is also used in the sample buffer 220 to select the appropriate 8-times oversampled input data to enter the multiplier 210.

Figure 12 is a schematic diagram showing an implementation of the DASRC shown in Figure 1, in which the functions of the delay adjuster filter 20 and the clock processor 30 are performed by a single digital signal processor (DSP) 50 such as a Motorola DSP56000 integrated circuit. The DSP 50 is linked to the eight-times oversampler 10 and the two-times downsampler 40 by an application specific integrated circuit (ASIC) or logic cell array (LCA) 60.

In the DASRC shown in Figure 12, successive input samples are supplied in encoded form to an Audio Engineering Society / European Broadcasting Union Standard (AES/EBU) decoder 15, which supplies decoded input samples at a sampling frequency of fs_{in} to the eight-times oversampler 10 and also supplies an input interrupt signal 25 to the DSP 50. The input interrupt signal 25 has a frequency of $8fs_{in}$ and is generated as part of the operation of the AES/EBU decoder 15.

After the input samples have been oversampled by the eight-times oversampler 10 they are passed to the ASIC/LCA 60 which acts as an interface for the passing of samples to and from the DSP 50. The oversampled output samples (at a sampling frequency of $2fs_{out}$) generated by the DSP 50 are passed, via the ASIC/LCA 60, to the two-times downsampler 40 in which they are converted (e.g. by filtering accompanied by the discarding of alternate samples) to output samples at a sampling frequency of fs_{out}. The output samples are passed to an AES/EBU encoder 35 for encoding for transmission to other apparatus (not shown).

A clocking signal at fs_{out} is received by the apparatus so that the sampling frequency of the output samples can be synchronised with that used in other apparatus. The clocking signal is passed to a multiplier 45 (which may be embodied as a second AES/EBU decoder) which

generates an output interrupt signal 55 having a frequency of $2fs_{out}$ from the clocking signal. The output interrupt signal 55 is passed to the DSP 50.

The input interrupt signal 25 and the output interrupt signal 55 are used to control the input of samples to the DSP 50 and the output of samples from the DSP 50 respectively. The input interrupt signal 25 is processed by the DSP 50 relatively quickly and with a higher interrupt priority than (i.e. in preference to) the output interrupt signal 55. This is achieved in the present embodiment by arranging for the input interrupt signal to use the 'short interrupt' mechanism of the Motorola DSP56000 integrated circuit referred to above. Interrupt signals processed by this mechanism are executed quickly and with high priority.

The main calculations carried out by the DSP 50 (that of the clock processor 30 and the delay adjuster filter 20) are performed using programmable on-chip registers of the Motorola DSP56000 integrated circuit. The operation of the DSP 50 is controlled by a number of programmed computation routines stored in the DSP, with each of the computation routines corresponding to one of the main functions of the DASRC such as calculation of the values $Ts_{out}(est)$ and dt(est) or the generation of coefficient addresses to be supplied to the coefficient store 200. The computation routines are initiated by the DSP 50 in response to the interrupt signals 25 and 55. In this way, the single DSP 50 controls the overall operation of the DASRC.

CLAIMS

1. A multi-tap oversampling filter for oversampling input data by an oversampling factor of m x n, m and n being integers, the filter coefficients of the multi-tap oversampling filter being defined by a convolution of the coefficients of a first oversampling filter having an oversampling factor of m with the coefficients of a second oversampling filter,

the second oversampling filter having an oversampling factor of n and having m-1 fixed-value coefficients inserted between adjacent filter coefficients.

2. A multi-tap oversampling filter according to claim 1, in which the fixed-value coefficients are zero-valued coefficients.

3. A sampling frequency converter for operating on sampled input data having an input sampling frequency to generate sampled output data having an output sampling frequency, the sampling frequency converter comprising:

a multi-tap oversampling filter according to claim 1 or claim 2 for operating on the input data to generate oversampled input data, the oversampled input data having a sampling frequency greater than the output sampling frequency; and

a downsampler for downsampling the oversampled input data to generate the output data.

4. A sampling frequency converter according to claim 3, in which the downsampler comprises means for selecting a respective sample of the oversampled input data for use as each sample of the output data, the selected sample of the oversampled input data being the sample temporally closest to that sample of the output data.

5. A sampling frequency converter according to claim 4, in which samples of the oversampled input data not selected by the downsampler are not generated.

6. A method of generating filter coefficients for a multi-tap

oversampling filter for oversampling input data by an oversampling factor of m x n, m and n being integers, the method comprising the steps of:

inserting m-1 fixed-value coefficients between adjacent filter coefficients of a first oversampling filter having an oversampling factor of n; and

convolving the coefficients of the first oversampling filter with the coefficients of a second oversampling filter having an oversampling factor of m.

7. A method according to claim 6, in which the fixed-value coefficients are zero-valued coefficients.

8. A method of increasing the oversampling factor of a multi-tap oversampling filter by a factor of n, where n is an integer, the method comprising the steps of:

inserting n-1 fixed-value coefficients between adjacent filter coefficients of the multi-tap oversampling filter; and

convolving the coefficients of the multi-tap oversampling filter with the coefficients of a second multi-tap oversampling filter having an oversampling factor of n.

9. A method according to claim 8, in which the fixed-value coefficients are zero-valued coefficients.

10. A multi-tap oversampling filter substantially as hereinbefore described with reference to the accompanying drawings.

11. A method of generating filter coefficients for a multi-tap oversampling filter, the method being substantially as hereinbefore described with reference to the accompanying drawings.

12. A method of increasing the oversampling factor of a multi-tap oversampling filter, the method being substantially as hereinbefore described with reference to the accompanying drawings.

Patents Act 1977
Examiner's report to the Comptroller under Section 17 (The Search Report)

Application number
GB 9210861.2

Relevant Technical fields

(i) UK Cl (Edition K) H3U (UGF)

(ii) Int Cl (Edition 5) H03H 17/06

Search Examiner
D MIDGLEY

Databases (see over)

(i) UK Patent Office

(ii) ONLINE DATABASES: WPI

Date of Search
30 SEPTEMBER 1992

Documents considered relevant following a search in respect of claims 1-9

Category (see over)	Identity of document and relevant passages	Relevant to claim(s)
A	US 4472785 (VICTOR) whole document	1,6,8

Category	Identity of document and relevant passages	Relevant to claim(s)

Categories of documents

X: Document indicating lack of novelty or of inventive step.

Y: Document indicating lack of inventive step if combined with one or more other documents of the same category.

A: Document indicating technological background and/or state of the art.

P: Document published on or after the declared priority date but before the filing date of the present application.

E: Patent document published on or after, but with priority date earlier than, the filing date of the present application.

&: Member of the same patent family, corresponding document.

Databases: The UK Patent Office database comprises classified collections of GB, EP, WO and US patent specifications as outlined periodically in the Official Journal (Patents). The on-line databases considered for search are also listed periodically in the Official Journal (Patents).

Published 1993 at The Patent Office, Concept House, Cardiff Road, Newport, Gwent NP9 1RH. Further copies may be obtained from Sales Branch, Unit 6, Nine Mile Point, Cwmfelinfach, Cross Keys, Newport, NP1 7HZ. Printed by Multiplex techniques ltd, St Mary Cray, Kent.